UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/541,266	06/30/2005	Jan Hoogerbrugge	NL02 1410 US	6282

65913 7590 03/21/2008
NXP, B.V.
NXP INTELLECTUAL PROPERTY DEPARTMENT
M/S41-SJ
1109 MCKAY DRIVE
SAN JOSE, CA 95131

EXAMINER	
TSAI, SHENG JEN	

ART UNIT	PAPER NUMBER
2186	

NOTIFICATION DATE	DELIVERY MODE
03/21/2008	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

ip.department.us@nxp.com

Office Action Summary	Application No.	Applicant(s)	
	10/541,266	HOOGERBRUGGE, JAN	
	Examiner	**Art Unit**	
	SHENG-JEN TSAI	2186	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *16 January 2008*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-15 and 17-26* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-15 and 17-26* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *30 June 2005* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This Office Action is taken in response to Applicant's Amendments and Remarks filed on January 16, 2008 regarding application 10/541,266 filed on June 30, 2005.

2. Claims 1, 14-15, 17, 19 and 26 have been amended.

Claim 16 has been cancelled.

Claims 1-15 and 17-26 are pending for consideration.

3. ***Response to Remarks and Amendments***

Applicant's amendments and remarks have been fully and carefully considered, with the Examiner's response set forth below.

(1) In view of the amendments, rejection of claim 19 under 35 U.S.C. 112, second paragraph, has been withdrawn.

(2) In view of the amendments, rejection of claim 26 under 35 U.S.C. 112, second paragraph, has been withdrawn.

(3) Applicant contends that it is improper to combine the Norman reference with the Margo reference, because the Norman reference uses flash memory that is block oriented while the Margo reference uses an address scheme that may address one, two, three or four bytes of data. The Examiner disagrees.

First, by definition, a "block" of data is simply a "collection of data that may be treated as a unit." As to the "size" of a block, it may vary from as small as one byte to as large as 1024 bytes. Thus, address scheme disclosed by Margo that may address one, two, three or four bytes of data would readily apply for blocks that comprising just a few bytes. Again, Applicant acknowledges that Margo's addressing scheme may address 4-

byte as a unit, thus a “block” comprising “4-byte” is directly addressable using the scheme.

Second, address scheme disclosed by Margo hat may address one, two, three or four bytes are also directly applicable to block-oriented flash memory disclosed by Norman. Here, instead of addressing “byte” as a unit, the “block” is treated as an addressable unit. For example, a “block” comprising 128 bytes may be treated, as far as addressing is concerned, as “one byte” and addressed accordingly; a “block” comprising 256 bytes may be treated, as far as addressing is concerned, as “two-byte” and addressed accordingly; and a “block” comprising 512 bytes may be treated, as far as addressing is concerned, as “four-byte” and addressed accordingly; and so on.

Thus, there is nothing improper combining the Norman and Margo references as alleged by Applicant.

Therefore, the Examiner’s position regarding the patentability of all claims remains the same as stated in the previous Office Action.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claims 1-5, 9-14, 17-20 and 24-26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Norman (US 6,438,665), and in view of Margo (US 6,151,658).

As to claim 1, Norman discloses **a controller** [the control engine, figure 3, 130; the system includes a controller which includes logic circuitry which performs the comparison (abstract); column 8, lines 8-18] **for a memory** [the flash memory cell array, figure 3, 216] **having at least one memory cell** [the invention is a memory system including one or more arrays of memory cells (e.g., flash memory cells or other nonvolatile memory cells) (column 7, lines 41-43); Flash memory chip 3 shown in FIG. 3 includes controller 129 and array 216 of nonvolatile memory cells (which are preferably nonvolatile flash memory cells) (column 8, lines 65-67)], **that involves a higher cost for writing than for reading** [This bypass of a write operation results in a big time savings, since flash writes are slow compared to reads. A flash write of a sector can take 1 to 5 milliseconds while a read compare will take approximately 50 microseconds. Thus, a large savings is gained when a sector is not required to be written to array 216 (column 10, lines 44-49); In flash memory systems, writes of data to flash memory cells are slow and they cause wear on the cells. This wear limits the useful life of conventional flash memory systems and reduces their overall reliability (column 7, lines 8-11)], **said memory cell being allocated a first address information and adapted to store memory data** [figure 3 shows that ADDRESS, DATA and CONTROL information are provided by a host processor (2); figure 3 also shows that the ADDRESS information is directed to the flash memory cell array 9216) via flash interface (114); Each cell is indicative of a stored data bit (column 7, line 48)], **said memory controller comprising**:

a register connected with said memory [the corresponding register is the flash interface (figure 3, 114), which is connected to the flash memory cell array (figure 3, 216); note that a buffer memory (figure 3, 104) is also provided to stored data from the host processor (figure 3, 2) that is to be written to the flash memory; Margo also teaches a register comprising a CAM address store (figure 2, 68) and a RAM data store (figure 2, 70)], **and comprising register space for write data and for address information allocated thereto** [figure 3 shows that ADDRESS and DATA and ECC information are being allocated to the flash interface; Flash interface 114 receives data (to be written to array 114) and address bits from other elements of chip 3 and asserts corresponding data and address bits with appropriate timing and format to array 216 (column 9, lines 17-21); note that a buffer memory (figure 3, 104) is also provided to stored data from the host processor (figure 3, 2) that is to be written to the flash memory; Margo also teaches a register comprising a CAM address store (figure 2, 68) which stores address information and a RAM data store (figure 2, 70) that stores data], **a write controller** [the control engine, figure 3, 130; Margo also teaches a write buffer controller (figure 2, 16)] **connected with said register and said memory** [as shown in figure 3; also see Margo, figure 2], **and adapted to receive a write request comprising said first address information and first write data allocated thereto** [figure 3 shows that ADDRESS, DATA and CONTROL information are provided by a host processor (2) are received by the control engine (130)],

ascertain whether said first address information is stored in said register

if yes, compare said first write data with second write data of an earlier write request in said register allocated to said first address information [this limitation is taught by Margo, see below],

if no, compare said first write data with said memory data allocated to the first address information forward said first address information and said first write data to said register, respectively [the system also includes circuitry which compares new data (to be written to a set of cells) with stored data (preread from a corresponding set of cells) and prevents a write of the new data to the array if the new data is identical to the stored data (abstract); the data and address are forward to the flash interface (figure 3, 114) and the flash memory (figure 3, 216) only if the comparator indicates a mismatch, as shown in figure 3; If the new data is not identical to the stored data, the system writes the new data to a second one of the sectors (or other sets) of the cells (column 7, lines 54-58); If a miscompare is detected, the controller will then generate the necessary control signals to write the new set of data in buffer memory 104 to a new (erased) sector of flash memory array 216 (column 10, lines 30-34); column 9 line 66 to column 10, line 45],

initiate a write operation of said first or second write data, respectively, from said register to said memory, if the first or second write data, respectively, is different from said memory data [The system prevents a write of the new data to the array if the new data is identical to the stored data. If the new data is not identical to the stored data, the system writes the new data to a second one of the sectors (or other sets) of the cells (column 7, lines 54-58); If a miscompare is detected, the controller will

then generate the necessary control signals to write the new set of data in buffer memory 104 to a new (erased) sector of flash memory array 216 (column 10, lines 30-34); column 9 line 66 to column 10, line 45].

Regarding claim 1, Norman does not teach ascertain whether said first address information is stored in said register, and if yes, compare said first write data with second write data of an earlier write request in said register allocated to said first address information.

However, Margo teaches this limitation in the invention “Write-Buffer FIFO Architecture with Random Access Snooping Capability.” Specifically, Margo teaches **a memory controller** [see figure 2] **comprising**:

a register connected with said memory [the corresponding register comprising a CAM address store (figure 2, 68) and a RAM data store (figure 2, 70); the memory is shown in figure 1 as the “consumer” (14)], **and comprising register space for write data and for address information allocated thereto** [a register comprising a CAM address store (figure 2, 68) which stores address information and a RAM data store (figure 2, 70) that stores data],

a write controller [the write buffer controller (figure 2, 16)] **connected with said register and said memory** [see figures 1 and 2], **and adapted to receive a write request comprising said first address information and first write data allocated thereto** [figure 1 shows that ADDRESS and DATA information are provided by a host producer (10) are received by the write buffer (12) and then forwarded to the memory (i.e., the consumer, 14)],

ascertain whether said first address information is stored in said register [A producer provides the address store with an input write address and provides the data store with input write data. <u>The content addressable memory concurrently compares the input write address to the addresses present in the address store</u>. If the input write address is "related" to an address present in the address store, the content addressable memory detects an address hit. The indication of an address hit is produced to the write buffer controller which signals the data store to store the input write data in the rank of the data store associated with the "related" address detected by the content addressable memory (column 2, lines 26-37)],

if yes, compare said first write data with second write data of an earlier write request in said register allocated to said first address information [If the input write address is "related" to an address present in the address store, the content addressable memory detects an address hit. The indication of an address hit is produced to the write buffer controller which signals the data store to store the input write data in the rank of the data store associated with the "related" address detected by the content addressable memory. If the input write data does not overlap the valid portion of the data previously stored in the rank associated with the "related" address, the store operation results in write merging. If the input write data overlaps the valid portion of the data previously stored in the rank associated with the rank associated with the "related" address, the store operation results in write collapsing. The write buffer thus eliminates the need to allocate a new rank to store write data when an input

write address is "related" to an address present in the address store (column 2, lines 30-46)].

Margo also teaches that the motivation to ascertain the first address information is in the register and to compare the first write data with the second write data allocated to the first address is to provide efficient storage of write data and to reduce the number of read cycles for retrieving write data for related address [By storing write data for "related" addresses in a single rank rather than multiple ranks, the write buffer provides efficient storage of write data for "related" addresses. By eliminating the use of multiple ranks for storing write data for "related" addresses, the write buffer also reduces the number of read cycles for retrieving write data for "related" addresses going to a consumer (column 2, lines 46-53)].

Therefore, it would have been obvious for one of ordinary skills in the art at the time of Applicant's invention to ascertain the first address information is in the register and to compare the first write data with the second write data allocated to the first address, as demonstrated by Margo, and to incorporate it into the existing scheme disclosed by Norman in order to provide efficient storage of write data and to reduce the number of read cycles for retrieving write data for related address.

As to claim 2, Norman in view of Margo teaches that **the memory controller of claim 1, wherein said register is a FIFO register** [Margo teaches that "a write buffer is conceptually a FIFO (First-In-First-Out) device where data is posted at a front end of the FIFO and is retrieved from a back end of the FIFO …" (column 1, lines 13-34)].

As to claim 3, Norman in view of Margo teaches that **the memory controller of claim 1, further comprising a read controller** [Margo: figure 2 shows a read controller comprising a read pointer (20) and a read multiplexer (24) performing the function of read merge (RD_MERGE) and providing read address (RD_ADDR)] **connected with said register and said memory** [see Margo, figure 2], **and adapted to receive a read request comprising said first address information ascertain whether said first address information is stored in said register forward said read request to said register or said memory, depending on whether or not, respectively, said first address information is stored in said register** [In addition, the content addressable memory detects whether an input read address provided by a producer to a consumer is "related" to an address in the address store. If the input read address is "related" to an address in the address store, the valid write data in the data store associated with the "related" address is merged with data from the consumer associated with the input read address, thereby resulting in read merging. The merged data preserves read data coherency by reflecting both the most recent write data in the write buffer and the read data in the consumer. With this read merge function, a write buffer need not be emptied before a read is performed from the consumer (column 2, lines 54-65)].

As to claim 4, Norman in view of Margo teaches that **the memory controller of claim 3, wherein said write controller is adapted to send to said read controller a read request upon reception of said write request, said read request comprising said first address information contained in said write request** [Norman: The

controller will then fetch a byte of data from flash memory array 216 and a corresponding byte of data from buffer memory 104. The corresponding bits of each byte will be … (column 9, line 66 to column 10, line 10); note that a “fetch” is a “read” operation; Flash interface 114 also reads data bits from cells of array 216 (e.g., from any selected sector of cells of array 216) and asserts the data bits which it reads with appropriate timing and format to other elements of chip 3 (including to comparator circuit 110) (column 9, lines 22-26)].

As to claim 5, Norman in view of Margo teaches that **the memory controller of claim 1, wherein the write controller is adapted to allocate a flag** [Norman: figure 6, 410, miscompare flag] **indicative of the result of the comparison to said first write data** [Norman: column 9 line 57 to column 10, line 36] **and to transfer the flag to said register along with said first write data and said first address information** [Norman: If a miscompare is detected, the controller will then generate the necessary control signals to write the new set of data in buffer memory 104 to a new (erased) sector of flash memory array 216. It will also mark the sector just read from flash memory array 216, which has been found to miscompare, to an "obsolete" state (column 10, lines 30-36; column 9 line 57 to column 10, line 36)], **and to initiate a write operation to the memory only for first write data for which the flag is indicative of a difference between said first write data and said memory data or said second write data, respectively** [Norman: the system also includes circuitry which compares new data (to be written to a set of cells) with stored data (preread from a corresponding set of cells) and prevents a write of the new data to the array if the

new data is identical to the stored data (abstract); the data and address are forward to the flash interface (figure 3, 114) and the flash memory (figure 3, 216) only if the comparator indicates a mismatch, as shown in figure 3; If the new data is not identical to the stored data, the system writes the new data to a second one of the sectors (or other sets) of the cells (column 7, lines 54-58)].

As to claim 9, Norman in view of Margo teaches that **the memory controller of claim 1, wherein the write controller is adapted to perform the comparison of said first write data with said second write data or said memory for at least one bit at a time** [Norman: figure 6, 400~406 shows that each bit is compared individually one at a time respectively].

As to claim 10, Norman in view of Margo teaches that **the memory controller of claim 9, wherein the write controller is adapted to perform the comparison byte by byte or bit by bit** [Norman: The controller will then fetch <u>a byte of data</u> from flash memory array 216 and <u>a corresponding byte of data</u> from buffer memory 104. The corresponding <u>bits</u> of each byte will be … (column 9, line 66 to column 10, line 10)].

As to claim 11, Norman in view of Margo teaches **the memory controller of claim 9, wherein the write controller is adapted to perform the comparison bit by bit** [Norman: figure 6, 400~406 shows that each bit is compared individually one at a time respectively; The controller will then fetch <u>a byte of data</u> from flash memory array 216 and <u>a corresponding byte of data</u> from buffer memory 104. The corresponding <u>bits</u> of each byte will be … (column 9, line 66 to column 10, line 10)].

As to claim 12, Norman in view of Margo teaches **the memory controller of claim 9, comprising an XOR-Gate with a first input for said first write data and a second input for said memory data or said second write data, respectively, and an output port connected with the write controller** [Norman: figure 6, 400~406; column 9 line 57 to column 10, line 36].

As to claim 13, Norman in view of Margo teaches **the memory controller of claim 9, wherein the write controller is adapted to forward said first address information and said first write data to said register after receiving at least one logical "TRUE"-signal from the output of the XOR-Gate** [Norman: figure 6, 400~406; The controller will then generate a clock sampling the compare condition, as determined by the logic level of the J input. If the J is low (the two bytes match each other), the JK register will remain clear indicating that the data has compared … (column 10, line 11-36); circuit 110 can add a "one" bit to the contents of the register each time a non-matching pair of data values is received at the "A" and "B" inputs of circuit 110 … (column 13, line 66 to column 14, line 16)].

As to claim 14, Norman in view of Margo teaches **a memory device comprising a memory with at least one non-volatile memory cell and a memory controller according to claims 1** [Norman: the flash memory cell array (figure 3, 216), the control engine (figure 3, 130); the invention is a memory system including one or more arrays of memory cells (e.g., flash memory cells or other nonvolatile memory cells) (column 7, lines 41-43); Flash memory chip 3 shown in FIG. 3 includes controller 129 and array 216 of nonvolatile memory cells (which are preferably nonvolatile flash

memory cells) (column 8, lines 65-67); Margo: figures 1 and 2; the consumer (figure 1, 14) is the memory device and the random access snooping write buffer (figure 1, 12 and figure 2) is the controller].

As to claim 17, Norman discloses **a method for writing to a non-volatile memory using a write queue,** [the flash memory cell array, figure 3, 216; abstract; figure 3, 104 shows a buffer memory for storing data to be written into the flash memory cell array (figure 3, 216); Margo also teaches using a write queue comprising a CAM address store (figure 2, 68) and a RAM data store (figure 2, 70)] **for a memory** [the flash memory cell array, figure 3, 216] **said memory comprising at least one memory cell for storing memory data** [the invention is a memory system including <u>one or more arrays of memory cells</u> (e.g., flash memory cells or other <u>nonvolatile memory cells</u>) (column 7, lines 41-43); Flash memory chip 3 shown in FIG. 3 includes controller 129 and array 216 of nonvolatile memory cells (which are preferably nonvolatile flash memory cells) (column 8, lines 65-67); Each cell is indicative of a stored data bit (column 7, line 48], **said memory cell being uniquely allocated at least a first memory address** [the invention is a memory system including <u>one or more arrays of memory cells</u> (e.g., flash memory cells or other <u>nonvolatile memory cells</u>) (column 7, lines 41-43); Flash memory chip 3 shown in FIG. 3 includes controller 129 and array 216 of nonvolatile memory cells (which are preferably nonvolatile flash memory cells) (column 8, lines 65-67)] **and requires a higher cost for writing than for reading** [This bypass of a write operation results in a big time savings, since <u>flash writes are slow compared to reads</u>. A flash write of a sector can take 1 to 5

milliseconds while a read compare will take approximately 50 microseconds. Thus, a large savings is gained when a sector is not required to be written to array 216 (column 10, lines 44-49); In flash memory systems, writes of data to flash memory cells are slow and they cause wear on the cells. This wear limits the useful life of conventional flash memory systems and reduces their overall reliability (column 7, lines 8-11)],

comprising the steps of:

receiving a write request comprising said first address information and first write data allocated thereto [figure 3 shows that ADDRESS, DATA and CONTROL information are provided by a host processor (2) are received by the control engine (130); figure 3 shows that ADDRESS and DATA and ECC information are being allocated to the flash interface; Flash interface 114 receives data (to be written to array 114) and address bits from other elements of chip 3 and asserts corresponding data and address bits with appropriate timing and format to array 216 (column 9, lines 17-21)],

ascertain whether said first address information is stored in said writing queue if yes, compare said first write data with second write data of an earlier write request in said register allocated to said first address information [this limitation is taught by Margo, see below],

if no, compare said first write data with said memory data allocated to the first address information forward said first address information and said first write data to said register, respectively [the system also includes circuitry which compares new data (to be written to a set of cells) with stored data (preread from a corresponding

set of cells) and prevents a write of the new data to the array if the new data is identical to the stored data (abstract); the data and address are forward to the flash interface (figure 3, 114) and the flash memory (figure 3, 216) only if the comparator indicates a mismatch, as shown in figure 3; If the new data is not identical to the stored data, the system writes the new data to a second one of the sectors (or other sets) of the cells (column 7, lines 54-58); If a miscompare is detected, the controller will then generate the necessary control signals to write the new set of data in buffer memory 104 to a new (erased) sector of flash memory array 216 (column 10, lines 30-34); column 9 line 66 to column 10, line 45],

forwarding said first address information and said first write data to said writing queue if the first write data is different from the second write data or said memory data, respectively [The system prevents a write of the new data to the array if the new data is identical to the stored data. If the new data is not identical to the stored data, the system writes the new data to a second one of the sectors (or other sets) of the cells (column 7, lines 54-58); If a miscompare is detected, the controller will then generate the necessary control signals to write the new set of data in buffer memory 104 to a new (erased) sector of flash memory array 216 (column 10, lines 30-34); column 9 line 66 to column 10, line 45; Margo also teaches this aspect: If the input write data does not overlap the valid portion of the data previously stored in the rank associated with the "related" address, the store operation results in write merging. If the input write data overlaps the valid portion of the data previously stored in the rank associated with the rank associated with the "related" address, the store operation

results in write collapsing. The write buffer thus eliminates the need to allocate a new rank to store write data when an input write address is "related" to an address present in the address store (column 2, lines 30-46)]**, and**

writing said first write data from said writing queue to said memory cell corresponding to said first address [The system prevents a write of the new data to the array if the new data is identical to the stored data. If the new data is not identical to the stored data, the system writes the new data to a second one of the sectors (or other sets) of the cells (column 7, lines 54-58); If a miscompare is detected, the controller will then generate the necessary control signals to write the new set of data in buffer memory 104 to a new (erased) sector of flash memory array 216 (column 10, lines 30-34); column 9 line 66 to column 10, line 45].

Regarding claim 17, Norman does not teach ascertain whether said first address information is stored in said register, and if yes, compare said first write data with second write data of an earlier write request in said register allocated to said first address information.

However, Margo teaches this limitation in the invention “Write-Buffer FIFO Architecture with Random Access Snooping Capability.” Specifically, Margo teaches **a memory controller** [see figure 2] **comprising**:

a writing queue connected with said memory [the corresponding writing queue comprising a CAM address store (figure 2, 68) and a RAM data store (figure 2, 70); the memory is shown in figure 1 as the “consumer” (14)], **and comprising register space for write data and for address information allocated thereto** [a register comprising

a CAM address store (figure 2, 68) which stores address information and a RAM data store (figure 2, 70) that stores data],

a write controller [the write buffer controller (figure 2, 16)] **connected with said writing queue and said memory** [see figures 1 and 2], **and adapted to receive a write request comprising said first address information and first write data allocated thereto** [figure 1 shows that ADDRESS and DATA information are provided by a host producer (10) are received by the write buffer (12) and then forwarded to the memory (i.e., the consumer, 14)],

ascertain whether said first address information is stored in said writing queue [A producer provides the address store with an input write address and provides the data store with input write data. <u>The content addressable memory concurrently compares the input write address to the addresses present in the address store</u>. If the input write address is "related" to an address present in the address store, the content addressable memory detects an address hit. The indication of an address hit is produced to the write buffer controller which signals the data store to store the input write data in the rank of the data store associated with the "related" address detected by the content addressable memory (column 2, lines 26-37)],

if yes, compare said first write data with second write data of an earlier write request in said writing queue allocated to said first address information [If the input write address is "related" to an address present in the address store, the content addressable memory detects an address hit. The indication of an address hit is produced to the write buffer controller which signals the data store to store the input

write data in the rank of the data store associated with the "related" address detected by the content addressable memory. If the input write data does not overlap the valid portion of the data previously stored in the rank associated with the "related" address, the store operation results in write merging. If the input write data overlaps the valid portion of the data previously stored in the rank associated with the rank associated with the "related" address, the store operation results in write collapsing. The write buffer thus eliminates the need to allocate a new rank to store write data when an input write address is "related" to an address present in the address store (column 2, lines 30-46)].

Margo also teaches that the motivation to ascertain the first address information is in the writing queue and to compare the first write data with the second write data allocated to the first address is to provide efficient storage of write data and to reduce the number of read cycles for retrieving write data for related address [By storing write data for "related" addresses in a single rank rather than multiple ranks, the write buffer provides efficient storage of write data for "related" addresses. By eliminating the use of multiple ranks for storing write data for "related" addresses, the write buffer also reduces the number of read cycles for retrieving write data for "related" addresses going to a consumer (column 2, lines 46-53)].

Therefore, it would have been obvious for one of ordinary skills in the art at the time of Applicant's invention to ascertain the first address information is in the writing queue and to compare the first write data with the second write data allocated to the first address, as demonstrated by Margo, and to incorporate it into the existing scheme

disclosed by Norman in order to provide efficient storage of write data and to reduce the number of read cycles for retrieving write data for related address.

As to claim 18, Norman in view of Margo teaches that **the method of claim 17, wherein said first write data of different write requests is written from said writing queue to said memory according to a FIFO rule** [Margo teaches that “a write buffer is conceptually a FIFO (First-In-First-Out) device where data is posted at a front end of the FIFO and is retrieved from a back end of the FIFO …” (column 1, lines 13-34)].

As to claim 19, Norman in view of Margo teaches that **the method of claim 17, further comprising the steps receiving a read request comprising said first address information** [Margo: figure 1 shows that the producer (10) issues a read request with read address RD_IN_ADDRESS (64) that is received by the random access snooping write buffer (12), which in turn forwards the read request to the consumer (14) with RD_OUT_ADDRESS (62); figure 2 shows a read controller comprising a read pointer (20) and a read multiplexer (24) performing the function of read merge (RD_MERGE) and providing read address (RD_ADDR)] **ascertain whether said first address information is stored in said writing queue** [In addition, the content addressable memory detects whether an input read address provided by a producer to a consumer is "related" to an address in the address store. If the input read address is "related" to an address in the address store, the valid write data in the data store associated with the "related" address is merged with data from the consumer associated with the input read address, thereby resulting in read merging.

The merged data preserves read data coherency by reflecting both the most recent write data in the write buffer and the read data in the consumer. With this read merge function, a write buffer need not be emptied before a read is performed from the consumer (column 2, lines 54-65)] **forwarding said read request to said writing queue or said memory, depending on whether or not, respectively, said first address information is stored in said writing queue** [In addition, the content addressable memory detects whether an input read address provided by a producer to a consumer is "related" to an address in the address store. If the input read address is "related" to an address in the address store, the valid write data in the data store associated with the "related" address is merged with data from the consumer associated with the input read address, thereby resulting in read merging. The merged data preserves read data coherency by reflecting both the most recent write data in the write buffer and the read data in the consumer. With this read merge function, a write buffer need not be emptied before a read is performed from the consumer (column 2, lines 54-65)],

receiving from said writing queue or said memory, respectively, said writing queue data or said memory data allocated to said first address information [In addition, the content addressable memory detects whether an input read address provided by a producer to a consumer is "related" to an address in the address store. If the input read address is "related" to an address in the address store, the valid write data in the data store associated with the "related" address is merged with data from the consumer associated with the input read address, thereby resulting in read

merging. The merged data preserves read data coherency by reflecting both the most recent write data in the write buffer and the read data in the consumer. With this read merge function, a write buffer need not be emptied before a read is performed from the consumer (column 2, lines 54-65)].

As to claim 20, Norman in view of Margo teaches that **the method of claim 17, comprising a step of performing a read request upon reception of said write request, said read request comprising said first address information contained in said write request** [Norman: The controller will then <u>fetch</u> a byte of data from flash memory array 216 and a corresponding byte of data from buffer memory 104. The corresponding bits of each byte will be … (column 9, line 66 to column 10, line 10); note that a “fetch” is a “read” operation; Flash interface 114 also reads data bits from cells of array 216 (e.g., from any selected sector of cells of array 216) and asserts the data bits which it reads with appropriate timing and format to other elements of chip 3 (including to comparator circuit 110) (column 9, lines 22-26)].

As to claim 24, Norman in view of Margo teaches that **the method of claim 17, comprising a step of comparing at least one bit at a time of said first write data and of said second write data or said memory data** [Norman: figure 6, 400~406 shows that each bit is compared individually one at a time respectively; The controller will then fetch <u>a byte of data</u> from flash memory array 216 and <u>a corresponding byte of data</u> from buffer memory 104. The corresponding <u>bits</u> of each byte will be … (column 9, line 66 to column 10, line 10)].

As to claim 25, Norman in view of Margo teaches that **the method of claim 24, wherein said comparing step comprises a step of performing an XOR-operation between said first write data and said memory data or said second write data, respectively** [Norman: figure 6, 400~406; column 9 line 57 to column 10, line 36].

As to claim 26, it recites substantially the same limitations as in claim 17, and is rejected for the same reasons set forth in the analysis of claim 17. Refer to “As to claim 17” presented earlier in this Office Action for details.

6. Claims 6-8 and 21-23 are rejected under 35 U.S.C. 103(a) as being unpatentable over Norman (US 6,438,665), in view of Margo (US 6,151,658), and further in view of Reams (US 6,438,660).

As to claim 6, Norman in view of Margo teaches the memory controller of claim 5 [see “As to claim 1” and “As to claim 5” presented earlier in this Office Action], but does not teach that the write controller is adapted to ascertain whether said register is full or empty.

Reams discloses in the invention “Method and Apparatus for Collapsing Writebacks to a Memory for Resource Efficiency” a scheme for reducing the number of writeback operations to memory by comparing the addresses of different writeback requests and collapse the requests if the addresses are the same [Method and apparatus are disclosed which increase resource efficiency by collapsing writebacks to a memory. In general the method and apparatus receive an address of a memory request and compare that address to addresses of writebacks stored in a memory controller in order to determine whether the memory request maps to the same

memory line of the memory as a stored writeback. If (1) the memory request generates a writeback, and (2) the memory request maps to the same line in the main memory as one of the stored writebacks, then the writeback generated from the memory request may be collapsed with one of the stored writebacks, thus reducing the number of writes to the main memory (abstract)].

Specifically, Reams teaches using a write buffer/queue [figure 2, 58] storing the addresses [figure 2, 68] and data [figure 2, 70] of the write requests, and the associated control logics [see figure 2], including a "<u>storage full monitor</u>" [figure 2, 60] which indicates if the buffer/queue is full [The storage full monitor 60 generates <u>a full signal</u> when the writeback storage 58 is full (i.e. has no storage elements 74 that are empty or available) … (column 11, lines 44-54)], and when storage locations are empty [The latched first address in step 422 is stored in <u>an empty location</u> in the address store 68 of the writeback storage 58 … (column 15, lines 6-15)].

Therefore, it would have been obvious for one of ordinary skills in the art at the time of Applicant's invention to include the "full" or "empty" indicators indicating the status of the write request buffer/queue, as demonstrated by Reams, and to incorporate it into the existing scheme of Norman in view of Margo, in order to prevent the buffer/queue from overflowing, which may cause request information being overwritten and lost, or from being underflowing, which may cause invalid requests to be executed.

As to claim 7, Norman in view of Margo and Reams teaches **the memory controller of claim 6, wherein the write controller is adapted to initiate at least**

one write operation from the register to the memory after assessing that the register is full [Reams: The storage full monitor 60 generates a full signal when the writeback storage 58 is full (i.e. has no storage elements 74 that are empty or available) … (column 11, lines 44-54); a write operation is performed when a pending MIC request is completed (column 11, line 44 to column 12, line 6); needs to perform one write operation when the register is full to prevent overflow of data at the register].

As to claim 8, Norman in view of Margo and Reams teaches **the memory controller claim 1, wherein the write controller is adapted to initiate a write operation from the register to the memory after assessing that the register is not empty and in case there is no pending write request** [Reams: The present example assumes that the writeback that was generated in response to the second read for ownership request of step 406 has yet to be written back to the second memory 18b and that no other pending writebacks are stored in the writeback storage 58. As a result, the comparator 56 generates the hit signal since the second address maps to the first memory line 28' and the writeback stored in the first storage element 74' maps to the first memory line 28' (column 16, lines 19-27); figures 3, 4A and 4B].

As to claim 21, it recites substantially the same limitations as in claim 6, and is rejected for the same reasons set forth in the analysis of claim 6. Refer to “As to claim 6” presented earlier in this Office Action for details.

As to claim 22, it recites substantially the same limitations as in claim 7, and is rejected for the same reasons set forth in the analysis of claim 7. Refer to “As to claim 7” presented earlier in this Office Action for details.

As to claim 23, it recites substantially the same limitations as in claim 8, and is rejected for the same reasons set forth in the analysis of claim 8. Refer to "As to claim 8" presented earlier in this Office Action for details.

7. Claim 15 are rejected under 35 U.S.C. 103(a) as being unpatentable over Norman (US 6,438,665), in view of Margo (US 6,151,658), and further in view of Sunaga et al. (US 6,785,154, hereinafter referred to as Sunaga).

As to claim 15, Norman in view of Margo teaches **a memory device according to claim 1** [see "As to claim 1" presented earlier in this Office Action], but does not teach that **the memory comprises memory cells from the group of MRAM and FERAM**.

However, the scheme disclosed by Norman in view of Margo is applicable to any type of memory cells, including from the group of MRAM and FERAM.

Further, Sunaga discloses **a memory device** [a magnetic random access memory (MRAM) circuit block … (abstract); figures 1 and 4], **wherein said memory comprises memory cells from the group of MRAM and FERAM** [a magnetic random access memory (MRAM) circuit block … (abstract); As shown in FIG. 3, the MTJ element 44 consists of at least three layers, i.e., a free layer 46, which is a <u>ferromagnetic layer</u> the magnetization direction of which can be changed, a tunneling barrier 48, which is an insulator layer conducting a tunnel current, and a pinned layer 50, which is a <u>ferromagnetic layer</u> having a fixed magnetization direction (column 1, lines 19-24)].

Therefore, it would have been obvious for one of ordinary skills in the art at the time of Applicant's invention to recognize that memory cells from the group of MRAM and FERAM are well known and commonly used in the art, as demonstrated by Sunaga, hence lacking patentable significance.

8. ***Related Prior Art***

The following list of prior art is considered to be pertinent to applicant's invention, but not relied upon for claim analysis conducted above.

- Abato et al., (US 5,627,993), "Methods and Systems for Merging Data during Cache Checking and Write-Back Cycles for Memory Reads and Writes."
- Rosich, (US 5,517,660), "Read-Write Buffer for Gathering Write Requests and Resolving Read Conflicts Based on a Generated Byte Mask Code."
- Margo, (US 6,678,838), "Method to Track Master Contribution Information in a Write Buffer."

Conclusion

9. Claims 1-15 and 17-26 are rejected as explained above.

10. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of

the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

11. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Sheng-Jen Tsai whose telephone number is 571-272-4244. The examiner can normally be reached on 8:30 - 5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Matthew Kim can be reached on 571-272-4182. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Sheng-Jen Tsai/
Partial Signatory Examiner, Art Unit 2186

March 11, 2008

/Pierre-Michel Bataille/
Primary Examiner, Art Unit 2186